

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the above-referenced application.

Listing of Claims:

1. (Previously Presented) A method of switching during run time from a first scheduler to a second scheduler for a multitasking system for a processor, comprising:

choosing the second scheduler from a plurality of schedulers, wherein at least one of the plurality of schedulers selects processes to be run from a plurality of runnable processes different from the plurality of schedulers and wherein choosing the second scheduler is based on parameters that vary according to run time conditions;

setting, during a context switch operation, a program counter to an address corresponding to code of the second scheduler; and

the processor executing code of the second scheduler at an address corresponding to the program counter.

2. (Previously Presented) A method, according to claim 1, further comprising:

setting a stack pointer to an address corresponding to stack space for the second scheduler; and

the processor using the stack space at the stack pointer after executing code at the address corresponding to the program counter.

3. (Original) A method, according to claim 1, wherein all of the schedulers use the same stack.

4. (Cancelled)

5. (Previously Presented) A method, according to claim 1, wherein at least one of the schedulers is for statistical code profiling.

6. (Previously Presented) A method, according to claim 1, wherein the first scheduler is for start up conditions and the second scheduler is for steady state operation.

7. (Previously Presented) A method, according to claim 1, wherein choosing the second scheduler is performed by setting up a return from an exception that causes the scheduler to execute.

8. (Previously Presented) A method, according to claim 1, wherein setting a program counter includes modifying a variable that is modified according to the second scheduler.

9. (Previously Presented) A method of scheduling tasks in a multitasking operating system, comprising:

using a first scheduler to schedule tasks;

choosing a second scheduler from a plurality of schedulers, wherein at least one of the plurality of schedulers selects processes to be run from a plurality of runnable processes different from the plurality of schedulers and wherein choosing the second scheduler is based on parameters that vary according to run time conditions; and

switching, during run time, from using the first scheduler to schedule tasks to using the second scheduler to schedule tasks.

10. (Previously Presented) A method, according to claim 9, wherein choosing the second scheduler is performed by setting up a return from an exception that causes the second scheduler to execute.

11. (Previously Presented) A method, according to claim 9, wherein switching to using the second scheduler includes setting a program counter to an address corresponding to code of the second scheduler.

12. (Previously Presented) A method, according to claim 11, wherein setting a program counter includes modifying a variable that is modified according to the second scheduler.

13. (Previously Presented) A method, according to claim 9, further comprising:

setting a stack pointer to an address corresponding to stack space for the second scheduler; and

the processor using the stack space at the stack pointer after executing code at the address corresponding to the program counter.

14. (Original) A method, according to claim 9, wherein all of the schedulers use the same stack.

15. (Cancelled)

16. (Previously Presented) A method, according to claim 9, wherein at least one of the schedulers is for statistical code profiling.

17. (Previously Presented) A method, according to claim 9, wherein the first scheduler is for start up conditions and the second scheduler is for steady state operation.

18. (Previously Presented) Computer software in combination with a computer readable medium that switches, during run time, from a first scheduler to a second scheduler for a multitasking system for a processor, comprising:

executable code, provided on a computer readable medium, that chooses the second scheduler from a plurality of schedulers, wherein at least one of the plurality of schedulers selects processes to be run from a plurality of runnable processes different from the plurality of schedulers and wherein executable code that chooses the second scheduler uses parameters that vary according to run time conditions;

executable code, provided on a computer readable medium, that sets a program counter to an address corresponding to code of the second scheduler; and

executable code, provided on a computer readable medium, that causes the processor to execute code at an address corresponding to the program counter.

19. (Previously Presented) Computer software, according to claim 18, further comprising:

executable code, provided on a computer readable medium, that sets a stack pointer to an address corresponding to stack space for the second scheduler; and

executable code, provided on a computer readable medium, that causes the processor to use the stack space at the stack pointer after executing code at the address corresponding to the program counter.

20. (Previously Presented) Computer software, according to claim 18, wherein all of the schedulers use the same stack.

21. (Cancelled)

22. (Previously Presented) Computer software, according to claim 18, wherein at least one of the schedulers is for statistical code profiling.

23. (Previously Presented) Computer software, according to claim 18, wherein the first scheduler is for start up conditions and the second scheduler is for steady state operation.

24. (Previously Presented) Computer software, according to claim 18, wherein executable code that causes the processor to execute code at an address sets up a return from an exception that causes the second scheduler to execute.

25. (Previously Presented) Computer software, according to claim 18, wherein executable code that sets a program counter modifies a variable according to the second scheduler.

26. (Previously Presented) Computer software in combination with a computer readable medium that schedules tasks in a multitasking operating system, comprising:

executable code, provided on a computer readable medium, that uses a first scheduler to schedule tasks;

executable code, provided on a computer readable medium, that chooses a second scheduler from a plurality of schedulers, wherein at least one of the plurality of schedulers selects processes to be run from a plurality of runnable processes different from the plurality of schedulers and wherein executable code that chooses the second scheduler uses parameters that vary according to run time conditions; and

executable code, provided on a computer readable medium, that switches, during run time, from using the first scheduler to schedule tasks to using the second scheduler to schedule tasks.

27. (Previously Presented) Computer software, according to claim 26, wherein executable code that chooses the second scheduler sets up a return from an exception that causes the scheduler to execute.

28. (Previously Presented) Computer software, according to claim 26, wherein executable code that switches to using the second scheduler sets a program counter to an address corresponding to code of the second scheduler.

29. (Previously Presented) Computer software, according to claim 28, wherein setting a program counter includes modifying a variable that is modified according to the second scheduler.

30. (Previously Presented) Computer software, according to claim 26, further comprising:

executable code, provided on a computer readable medium, that sets a stack pointer to an address corresponding to stack space for the second scheduler; and

executable code, provided on a computer readable medium, that causes the processor to use the stack space at the stack pointer after executing code at the address corresponding to the program counter.

31. (Previously Presented) Computer software, according to claim 26, wherein all of the schedulers use the same stack.

32. (Cancelled)

33. (Previously Presented) Computer software, according to claim 26, wherein at least one of the schedulers is for statistical code profiling.

34. (Previously Presented) Computer software, according to claim 26, wherein the first scheduler is for start up conditions and the second scheduler is for steady state operation.